

This document provides a comprehensive overview of the technologies used in the "Code Forge" club management application, along with a detailed breakdown of each page's purpose and features.

1. Overall Tech Stack

The "Code Forge" application is built as a **full-stack web application** using a combination of backend and frontend technologies:

- **Backend Framework:** Flask (Python)
- **Database:** SQLite (for local development) and PostgreSQL (for scalability and production environments)
- **Real-time Communication:** Flask-SocketIO (for chat)
- **Frontend (Structure):** HTML5
- **Frontend (Styling):** Tailwind CSS (with custom CSS for animations)
- **Frontend (Interactivity):** JavaScript (Vanilla JS)
- **Deployment (Assumed):** WSGI server (e.g., Gunicorn) for production, Flask's built-in server for development.

2. Library Usage

2.1 Python Libraries (Backend)

- **flask:** The micro-framework used to build the web application. It handles routing, requests, responses, sessions, and templating.
- **psycopg2:** The PostgreSQL adapter for Python. It allows the Flask application to connect to and interact with the PostgreSQL database.
 - **Usage:** Used in `get_db_connection()` for establishing database connections and in all `cur.execute()` calls for running SQL queries when PostgreSQL is the active database.
- **sqlite3:** Python's built-in SQLite database adapter.
 - **Usage:** (Implicitly used if the application is configured for SQLite in a development setup, although the provided `app.py` is now fully PostgreSQL-centric.)
- **werkzeug.security:** Provides utilities for password hashing and verification.
 - **Usage:** `generate_password_hash()` for storing user passwords securely in the database and `check_password_hash()` for verifying login credentials.
- **uuid:** Generates universally unique identifiers.
 - **Usage:** Used to create unique `user_ids` for students and mentors, and `room_ids` for brainstorming rooms. Also used for unique filenames for uploaded content.
- **smtplib:** Python's standard library for sending emails.

- **Usage:** Used in `send_otp()` function to send One-Time Passwords (OTPs) to users during registration.
- **random:** Generates random numbers.
 - **Usage:** Used to generate the 6-digit OTPs.
- **email.message.EmailMessage:** A class for creating email messages.
 - **Usage:** Used in `send_otp()` to construct the email content, subject, sender, and recipient.
- **os:** Provides functions for interacting with the operating system.
 - **Usage:** Used for managing file paths (e.g., `os.path.join`), creating directories (`os.makedirs`), and accessing environment variables (`os.environ.get`).
- **werkzeug.utils.secure_filename:** Secures a filename to prevent directory traversal vulnerabilities.
 - **Usage:** Used when saving uploaded files to ensure filenames are safe.
- **datetime:** Provides classes for working with dates and times.
 - **Usage:** Used for recording timestamps (e.g., `submitted_on`, `created_at`) and for comparing deadlines.
- **flask_socketio:** Integrates Socket.IO with Flask, enabling real-time, bidirectional communication between the client and server.
 - **Usage:** Powers the real-time chat and file sharing features in the brainstorming rooms. Handles `join`, `send_message`, and `share_file` events.
- **collections.defaultdict:** A dictionary subclass that calls a factory function to supply missing values.
 - **Usage:** Used in `mentor_dashboard` and `student_dashboard` to easily group event results by event title.
- **eventlet:** A concurrent networking library.
 - **Usage:** `eventlet.monkey_patch()` is used with Flask-SocketIO for asynchronous I/O, allowing the server to handle many concurrent connections (like chat messages) efficiently.

2.2 Frontend Libraries/Technologies

- **HTML5:** The standard markup language for creating web pages. All page structures are defined using HTML.
- **Tailwind CSS:** A utility-first CSS framework. It provides low-level utility classes that can be composed directly in HTML to build custom designs.
 - **Usage:** Used extensively for all styling, layout (flexbox, grid), responsiveness (`md:`, `lg:` prefixes), spacing, typography, colors, and basic visual components.
- **Font Awesome:** A popular icon library.
 - **Usage:** Provides various icons (e.g., `fas fa-lightbulb`, `fas fa-heart`, `fas fa-code`) used across the application for visual enhancement and clarity.
- **Vanilla JavaScript:** Plain JavaScript without any additional libraries or frameworks.

- **Usage:** Handles client-side interactivity, form submissions, dynamic content updates (e.g., QR code reveal, mobile menu toggle, typing animation), and real-time Socket.IO communication.

3. Application Sections and Pages

The application is structured into several distinct pages and sections, each serving a specific purpose:

3.1 Pre-Login Pages

- **Home Page (/)**
 - **Purpose:** The landing page for visitors before they log in. It provides an overview of the club and encourages joining.
 - **Sections:**
 - **Navigation Bar:** Club logo, "Join Us" button.
 - **Hero Section:** A welcoming message ("Unleash Your Coding Potential"), a motivational tagline, and a call to action to "Join Our Community" (linking to login). Features a bouncing robot image.
 - **Upcoming Events:** Displays a showcase of upcoming events with image, title, date, short description, and "Know More" button.
 - **Brainstorming Banner:** Attractively promotes the brainstorming feature, showcases example room ideas, and guides users on how to create rooms.
 - **Event Winners:** Shows a list of past event winners, grouped by event.
 - **Donation Banner:** A motivational quote about supporting the club, with a "Donate Now" button that reveals a QR code on click with an animation.
 - **About Us:** Briefly describes the "Code Forge" club, its vision, and mission.
 - **Footer:** Contains club name, quick links, and social media icons.
- **Login Page (/login)**
 - **Purpose:** Allows existing users (students, admins, mentors) to log into their respective dashboards.
 - **Features:** User ID and password input fields, "Log In" button, "Forgot Password?" link, and "Sign Up" link. Modern UI with animations.
- **Student Registration Step 1 (/register_student)**
 - **Purpose:** Collects initial student information and handles OTP verification for non-Marwadi University students.
 - **Features:** Fields for Name, College, Roll No, Email. OTP input field appears conditionally.
- **Student Registration Step 2 (/register_details)**
 - **Purpose:** Collects detailed student information after initial registration/OTP.
 - **Features:** Fields for Address, Contact, Year, Branch, Department, Password, Confirm Password.
- **Mentor Registration (/register_mentor)**

- **Purpose:** Allows mentors to register their accounts.
- **Features:** Fields for Name, College, Email, Expertise, Skills, Password, Confirm Password.

3.2 Student Dashboard & Related Pages

- **Student Dashboard (/student_dashboard)**
 - **Purpose:** The main hub for logged-in students, providing quick access to key club activities and information.
 - **Sections:**
 - **Navigation Bar:** Club logo, "Registered Events", "Brainstorming Ideas", "Profile", "Logout".
 - **Hero Section:** Personalized "Welcome, [Student Name]" with a dynamic typing effect on the name, User ID, and a motivational message. Features the robot image with a bounce effect.
 - **Upcoming Events:** (Same as Home Page) Displays event cards.
 - **Brainstorming Banner:** (Same as Home Page) Promotes brainstorming.
 - **Event Winners:** (Same as Home Page) Displays past winners.
 - **Donation Banner:** (Same as Home Page) Encourages donations with QR code.
 - **About Us:** (Same as Home Page) Club's vision and mission.
 - **Footer:** (Same as Home Page) Quick links and social media.
- **My Registered Events (/registered_events)**
 - **Purpose:** Shows a student all events they have registered for, along with detailed progress for each stage of those events.
 - **Features:** Event title, description, deadline. For each stage: stage title, deadline, submission status (Submitted/Not Submitted), submitted text (if any), download link for submitted file (if any), and submission date. A button to "Submit / View" for each stage.
- **Event Detail Page (/event/<int:event_id>)**
 - **Purpose:** Provides comprehensive details about a specific event.
 - **Features:** Event title, image (if available), full description, and deadline. Shows if the user is already registered and provides a "Register for Event" button or a "Go to Registered Events" button accordingly.
- **Submission Page (/submit/<int:event_id>/<int:stage_id>)**
 - **Purpose:** Allows students to submit their work for a specific stage of an event.
 - **Features:** Displays stage title. Input fields for submission text and a custom file upload area (PDF/PPT/PPTX) with a "+" icon and dashed border. "Submit" button. Includes a note about one-time submission.
- **Brainstorming Ideas Page (/brainstorm)**
 - **Purpose:** Allows students to create new brainstorming rooms or join existing ones.
 - **Features:** Form to "Create Room" with a title input. Lists "Active Rooms" with their titles and a "Join Room" button.
- **Brainstorm Room Page (/brainstorm/room/<room_id>)**

- **Purpose:** The real-time chat interface for a specific brainstorming session.
- **Features:** Displays room ID and admin name. Two main sections:
 - **Chat:** Real-time messages with usernames and timestamps.
 - **Shared Files:** A list of files shared within the room, with download links.
- **Interactive Elements (for non-mentors):** Message input field with "Send" button, and a file attachment/upload option with a "Share" button.
- **Student Profile Page (/profile)**
 - **Purpose:** Allows students to view their profile details and update certain information.
 - **Features:** Displays non-editable details like User ID, Name, Email, College, Roll No. Editable fields for Contact, Address, Year, Branch, Department with a "Save Changes" button. Separate section to "Change Password" with current, new, and confirm new password fields and a "Change Password" button. Flashed messages indicate success or failure of updates.

3.3 Admin Dashboard & Related Pages

- **Admin Dashboard (/admin_dashboard)**
 - **Purpose:** Central control panel for administrators to manage events and user data.
 - **Sections:**
 - **Navigation Bar:** Club logo, "View All Users" button, "Logout" button.
 - **Add New Event:** Form to create new events with fields for Title, Short Description, Full Description, Deadline, Event Image, and dynamic "Stages" (with title and deadline) that can be added.
 - **Announce Winners:** Form to announce top 3 winners for an event by entering event title, winner names, positions, and emails.
 - **All Events:** Lists all created events with their title, description, registration count, and submission count. Provides "Edit", "Delete", "View Submissions", and "View Progress" buttons for each event. "Delete" has a custom confirmation pop-up.
- **Per-Stage Progress (/view_progress/<int:event_id>)**
 - **Purpose:** Allows admins and mentors to track the submission progress of students for each stage of a specific event.
 - **Features:** Displays a table with student details (Name, Email, College, Roll No). Columns for each stage, showing submission status (Submitted/Not Submitted) and a "Download" link if a file was submitted.
- **All Registered Users (/view_all_users)**
 - **Purpose:** Provides administrators with a comprehensive list of all registered students and mentors in the system.
 - **Features:** A table displaying User ID, Name, Role (student/mentor), Email, Contact, Address, and College for all users.

3.4 Mentor Dashboard

- **Mentor Dashboard (/mentor_dashboard)**

- **Purpose:** Provides mentors with an overview of events they can oversee and access to brainstorming rooms and results.
- **Sections:**
 - **Navigation Bar:** Club logo, "Mentor Dashboard" label, "Logout" button.
 - **Welcome Message:** Personalized welcome with mentor's name and ID.
 - **Events:** Lists events with their title, description, deadline, stages, and a "View Progress" button (linking to [/view_progress](#)).
 - **Brainstorm Ideas:** Lists available brainstorming rooms with title, room ID, and a "View Room" button (linking to [/brainstorm/room](#)).
 - **Event Results:** Displays past event winners, grouped by event title (similar to student dashboard).
 - **Donation Banner:** (Same as student dashboard) Encourages support.
 - **About Us:** (Same as student dashboard) Club's vision and mission.
 - **Footer:** (Same as student dashboard) Quick links and social media.

This document should give you a complete understanding of your "Code Forge" application's structure, technologies, and features.